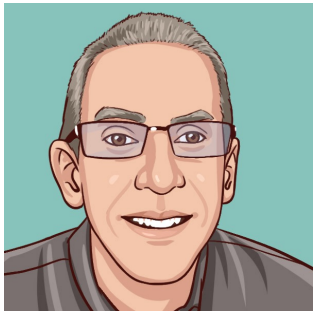

# Cheminformatics With Open-Source Software

**Pat Walters**

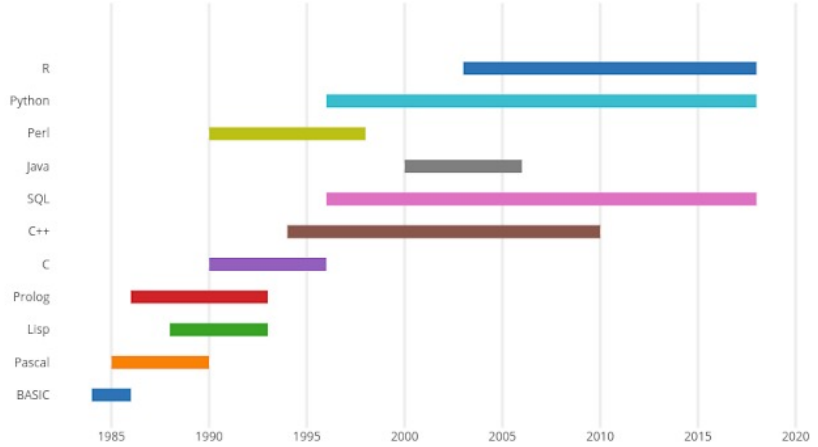
**August 19, 2021**

# About Me



**25 years in drug discovery**
**35 years writing software**
**A couple of years of blogging**

**Cheminformatics**
**Machine learning**





# About You

**A small amount of experience programming in Python**

**Knowledge of what organic molecules are**

**Curiosity and a desire to learn**

# What We're Going To Cover

**Exploratory data analysis**

- **A brief introduction to Jupyter notebooks**
- **A quick overview of the RDKit**
- **A lightning tour of the Pandas library for data analysis**

**A classification model example**

- **Build a decision tree**

**A regression model**

- **Predict aqueous solubility**

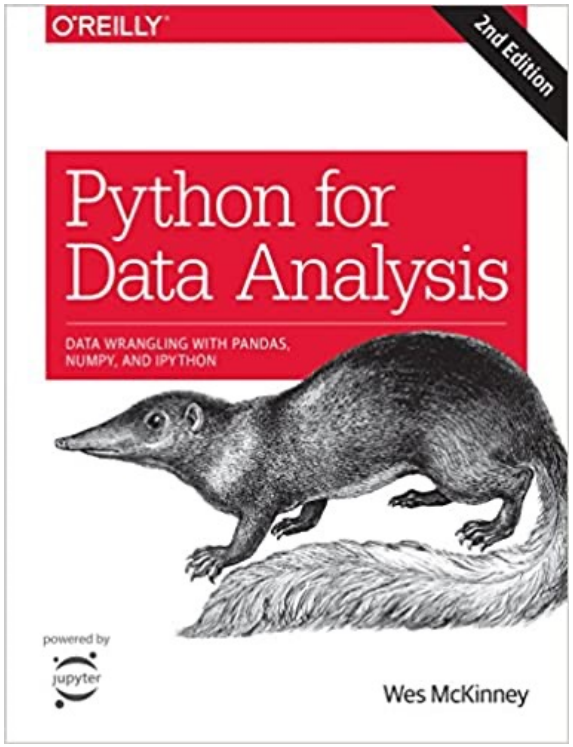
# What We're Not Going To Cover

**Neural networks**
**Learned molecular representations**
**Generative models**

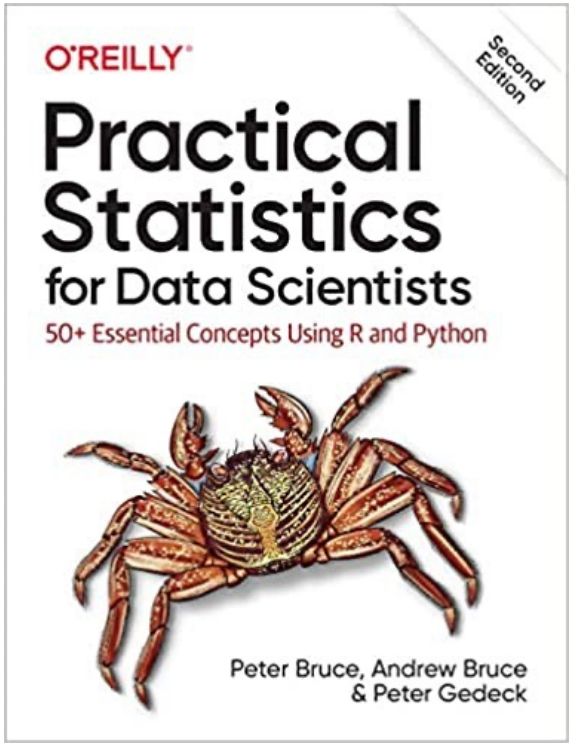
**Applicability domain**
**Model confidence**

**A lot theory**

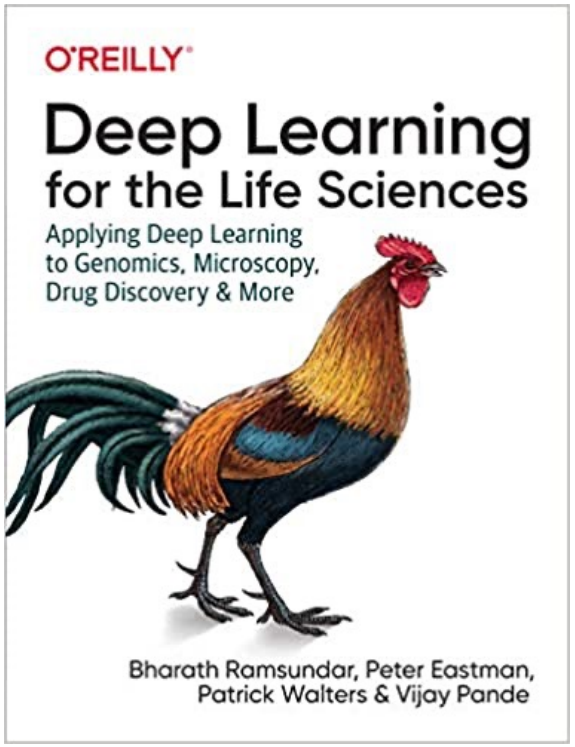
## Recommended Reading




https://www.amazon.com/Python-Data-Analysis-Wrangling-IPython/dp/1491957662/




https://www.amazon.com/Practical-Statistics-Data-Scientists-Essential/dp/149207294X/




https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837

# The RDKit Cookbook



## RDKit Cookbook

### Introduction

#### What is this?

This document provides example recipes of how to carry out particular tasks using the RDKit functionality from Python. The contents have been contributed by the RDKit community, tested with the latest RDKit release, and then compiled into this document. The RDKit Cookbook is written in reStructuredText, which supports Sphinx doctests, allowing for easier validation and maintenance of the RDKit Cookbook code examples, where appropriate.

#### What gets included?

The examples included come from various online sources such as blogs, shared gists, and the RDKit mailing lists. Generally, only minimal editing is added to the example code/notes for formatting consistency and to incorporate the doctests. We have made a conscious effort to appropriately credit the original source and authors. One of the first priorities of this document is to compile useful **short** examples shared on the RDKit mailing lists, as these can be difficult to discover. It will take some time, but we hope to expand this document into 100s of examples. As the document grows, it may make sense to prioritize examples included in the RDKit Cookbook based on community demand.

#### Feedback and Contributing

If you have suggestions for how to improve the Cookbook and/or examples you would like included, please contribute directly in the source document (the .rst file). Alternatively, you can also send Cookbook revisions and addition requests to the mailing list: <rdkit-discuss@lists.sourceforge.net> (you will need to subscribe first).

**Note**

The Index ID# (e.g., **RDKitCB_##**) is simply a way to track Cookbook entries and image file names. New Cookbook additions are sequentially index numbered, regardless of where they are placed within the document. As such, for reference, the next Cookbook entry is **RDKitCB_35**.

Open-Source Cheminformatics and Machine Learning

**Table of Contents**

RDKit Cookbook
- Introduction
  - What is this?
  - What gets included?
  - Feedback and Contributing
- Drawing Molecules (Jupyter)
  - Include an Atom Index
  - Include a Calculation
  - Include Stereo Annotations
  - Black and White Molecules
  - Highlight a Substructure in a Molecule
  - Without Implicit Hydrogens
  - With Abbreviations
- Bonds and Bonding
  - Hybridization Type and

https://www.rdkit.org/docs/Cookbook.html

# Useful Blogs

Practical Cheminformatics - This is a blog where I post once a month or so. These posts typically contain code that demonstrates various aspects of cheminformatics; clustering, machine learning, data visualization, etc. I occasionally throw in posts containing opinions on things like AI and getting a job.

Is Life Worth Living - A great blog from Iwatobipen (aka pen), whose posts are chock full of great code examples. Pen always seems to be up on the latest methods and posts interesting examples on a variety of topics ranging from quantum chemistry to machine learning.

The RDKit Blog - Greg Landrum is the primary contributor to, and BDFL, of the RDKit. In addition to the latest and greatest features in the RDKit, Greg's posts also touch on a number of key issues in Cheminformatics, such as dealing with unbalanced datasets and the impact of fingerprint folding on similarity searching.

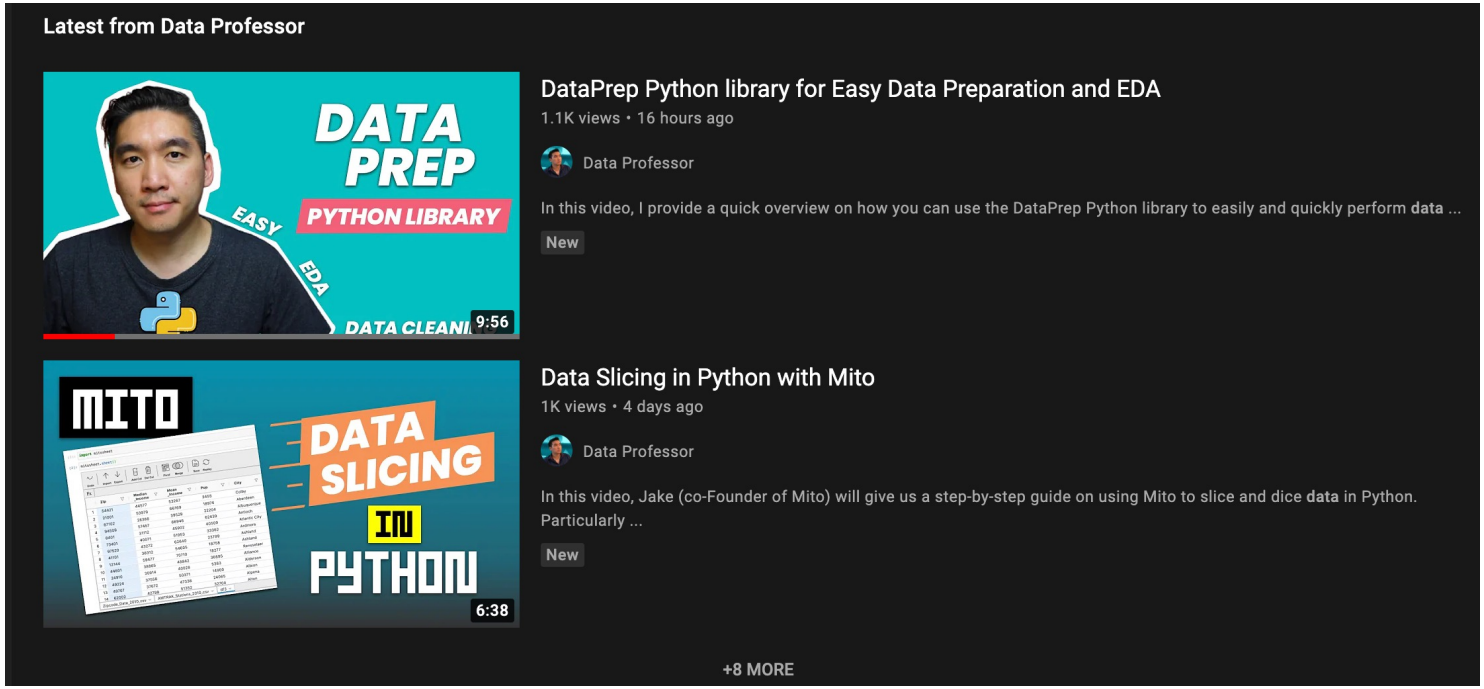
Reverie Labs Engineering Blog - The gang at Reverie Labs is doing some of the best work on applying machine learning in drug discovery. Their blog provides useful insights for those applying machine learning at scale.

The OpenBench Blog - A number of thoughtful posts on ADME modeling and the ways that we validate models.

Cheminformania - A set of very practical posts by Esben Jannik Bjerrum and friends that primarily focus on the applications of deep learning in drug discovery. These posts provide several useful code examples.

# YouTube

https://www.youtube.com/results?search_query=data+professor



## AI and “The Rise of the Machines”

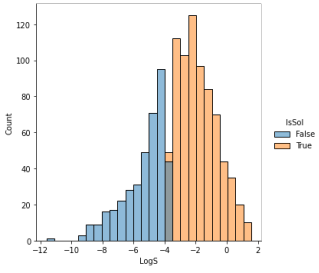
## What is Machine Learning?

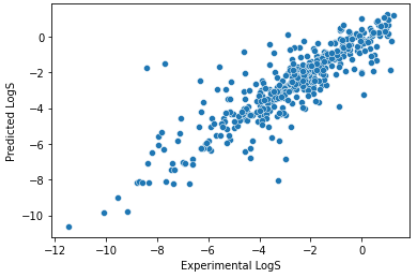
Machine learning is all about labeling things using examples.

**Cassie Kozyrkov, Google**

# Supervised vs Unsupervised Machine Learning

## Supervised ML

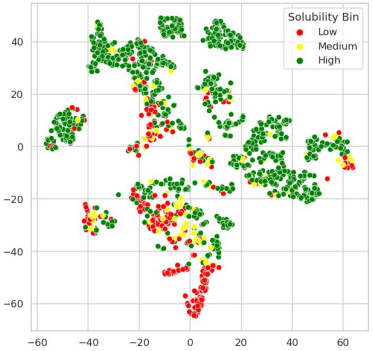


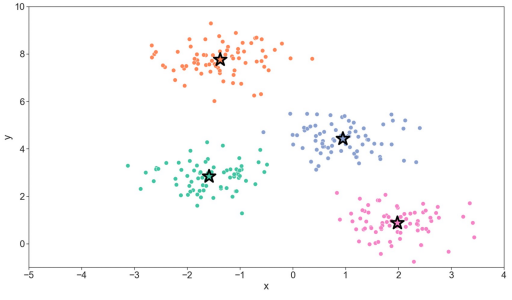




**Classification** **Regression**

**Predict**

## Unsupervised ML



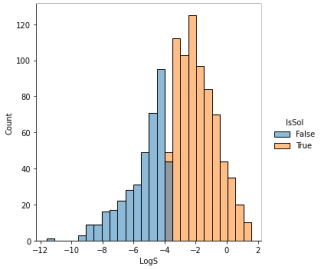



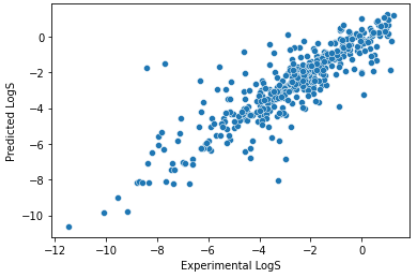

**Dimensionality Reduction** **Clustering**

**Group and Understand**

# Supervised vs Unsupervised Machine Learning

**Supervised ML**

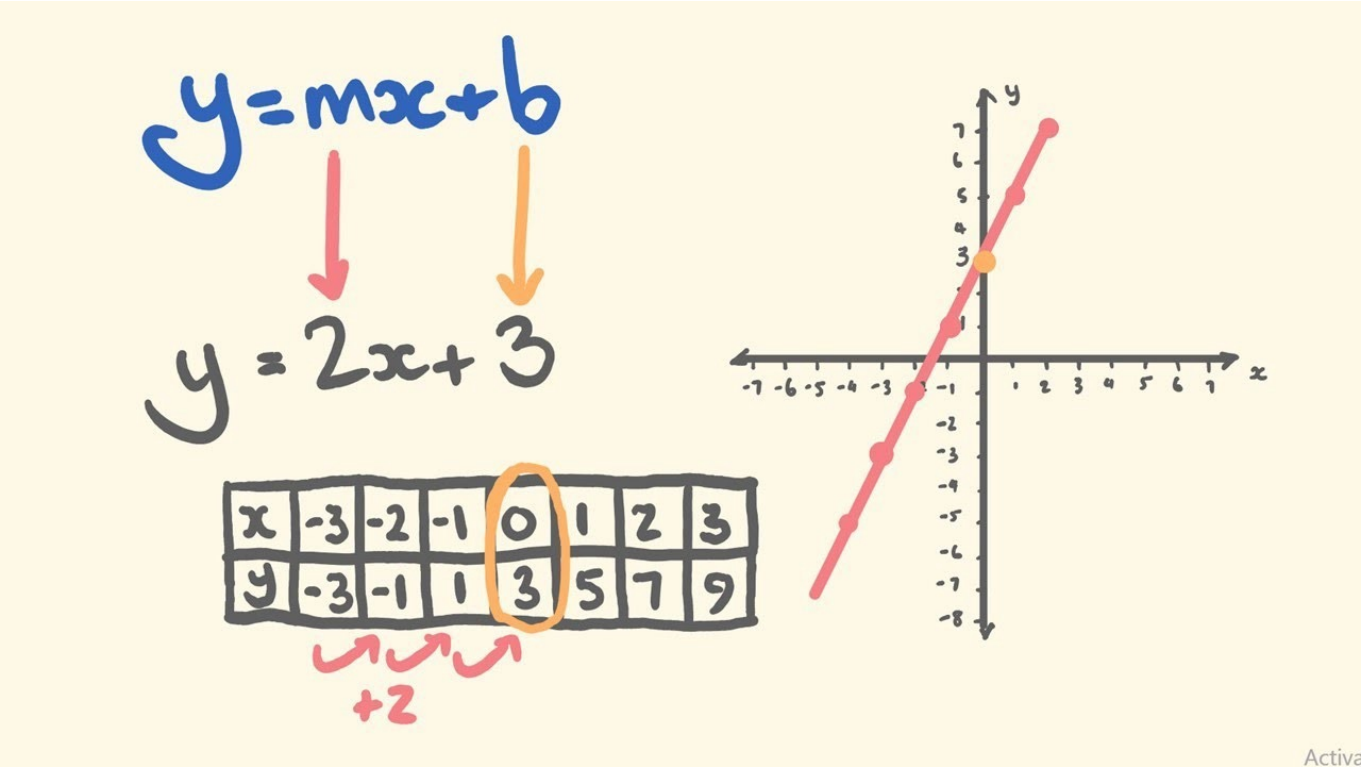



**Classification** **Regression**

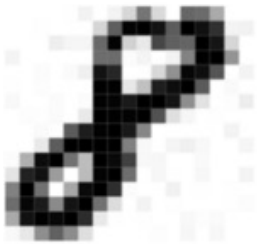
**Predict**

## What Are the Observables (X)? What is Being Predicted (Y)?



# Much of Computation is Based on Vector Representations

## Images



=


[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 107, 0, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 109, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 0, 253, 253, 23, 62, 224, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 8, 0, 1, 0, 0, 0, 0, 0, 4,
0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]


## Text

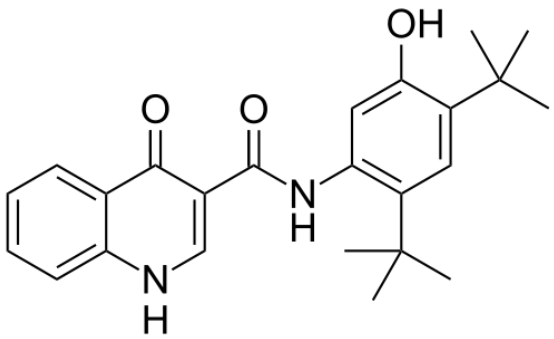
| Word | Index |
|---|---|
| It | 1 |
| was | 2 |
| the | 3 |
| best | 4 |
| of | 5 |
| times | 6 |
| worst | 7 |
| age | 8 |
| wisdom | 9 |
| foolishness | 10 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | it | was | the | best | of | times | worst | age | wisdom | foolishness |
| It was the worst of times | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| it was the best of times | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| it was the age of foolishness | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

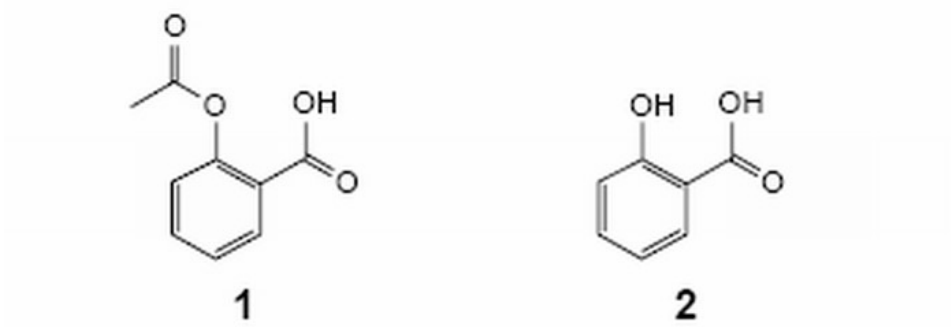
## Can We Create a Similar Vector Representation of a Molecule?

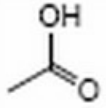
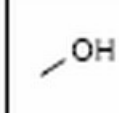
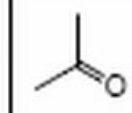
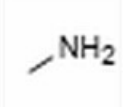
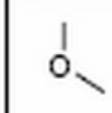
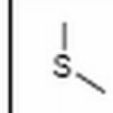
**Encode molecular features in computer readable format**



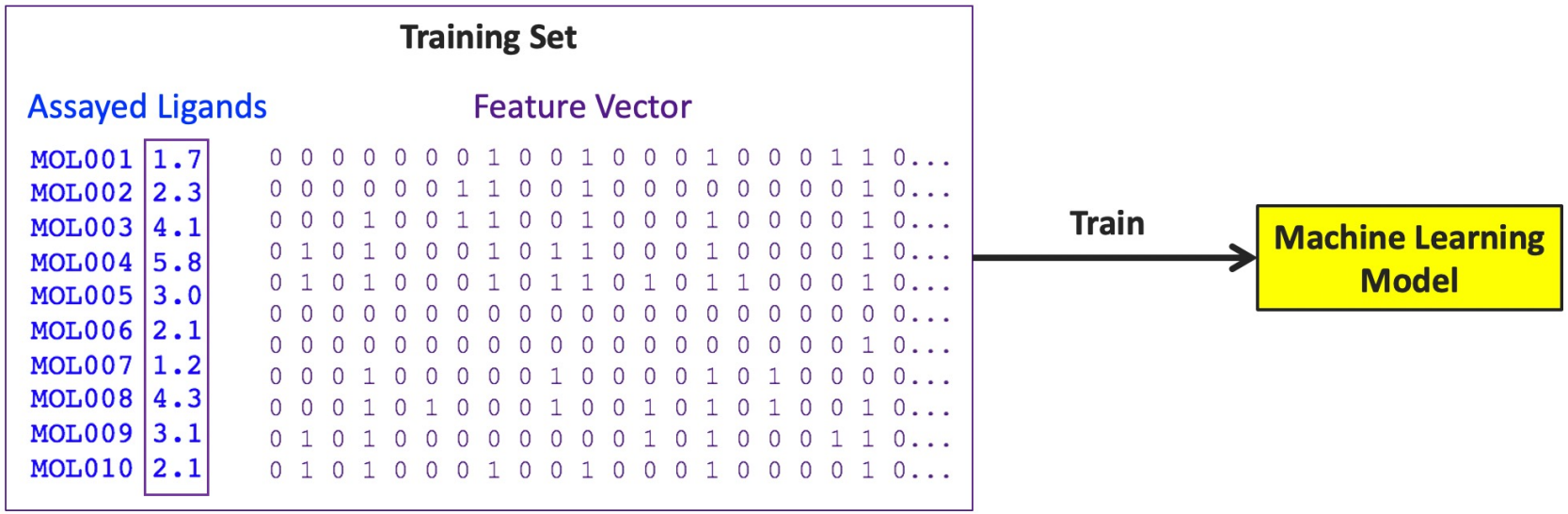

| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

## Chemical Fingerprints Represent Molecular Features

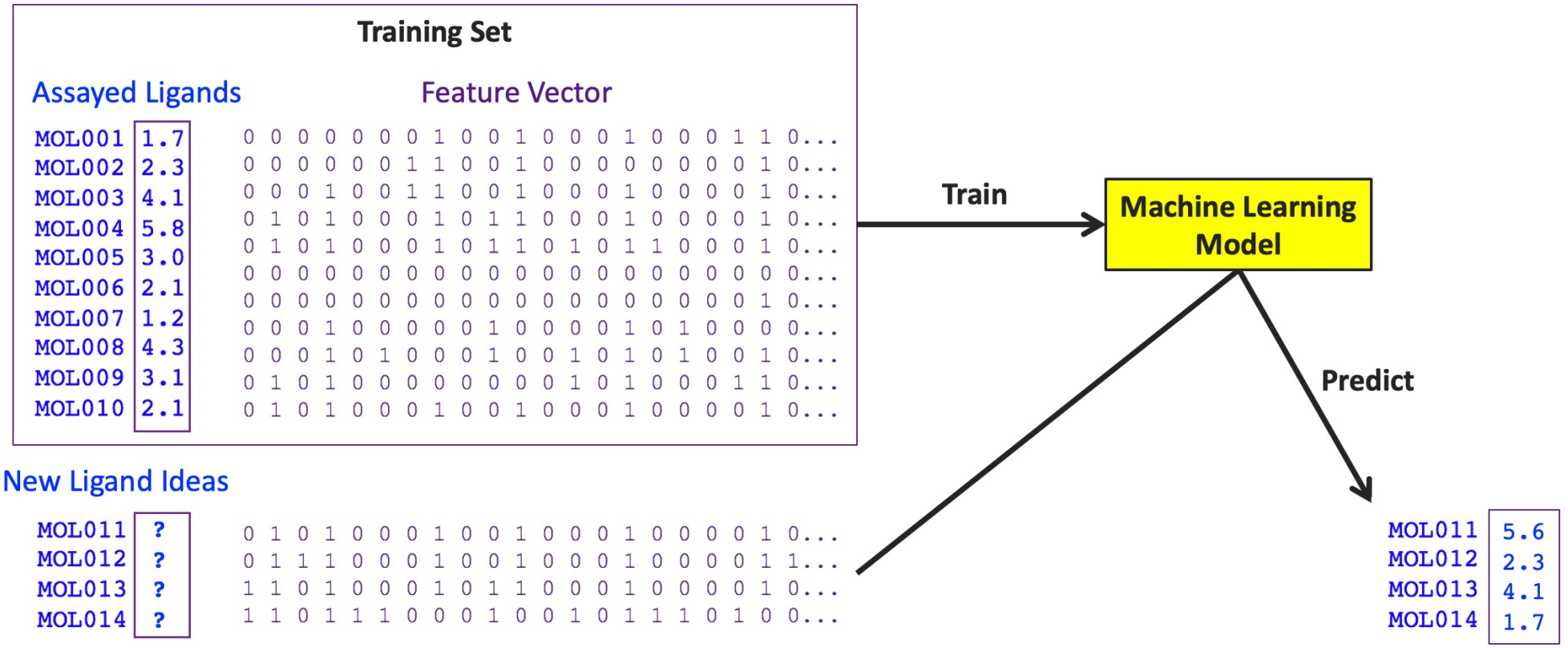



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| **2** | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| |  |  |  |  |  |  |  |  |

# Training a Machine Learning Model



# Making Predictions With a Machine Learning Model



## Important Jupyter Notebook Keyboard Shortcuts

| | |
|---|---|
| Esc-a | Insert cell above |
| Esc-b | Insert cell above |
| Esc-x | Delete cell(s) |
| Esc-m | Convert cell type to Markdown |
| Esc-y | Convert cell type to Code |

# Hands-on Part 1

# Building A Classification Model With SciKit Learn

```
train, test = train_test_split(df)
train_X = train[descriptors]
train_y = train.experiment
test_X = test[descriptors]
test_y = test.experiment
```

Construct training and test sets

```
my_model = RandomForestRegressor()
my_model.fit(train_X, train_y)
pred = my_model.predict(test_X)
```

Build a predictive model

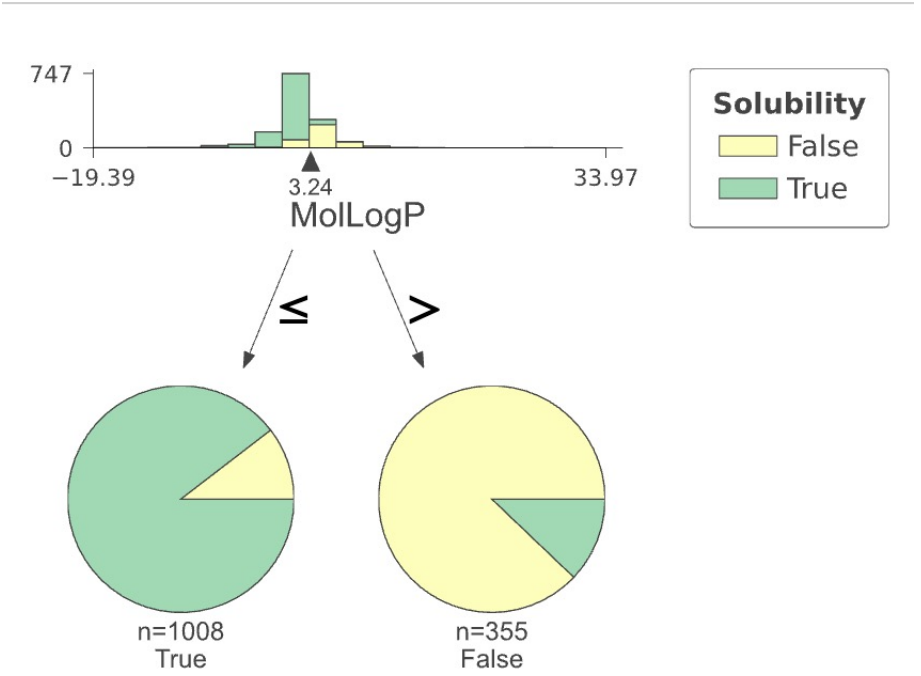
```
auc = roc_auc_score(test_y,pred)
mcc = matthews_corrcoef(test_y,pred)
kappa = cohen_kappa_score(test_y,pred)
```

Evaluate the predictive model

# The Decision Tree – A Simple ML Classifer

**Molecular Descriptors**

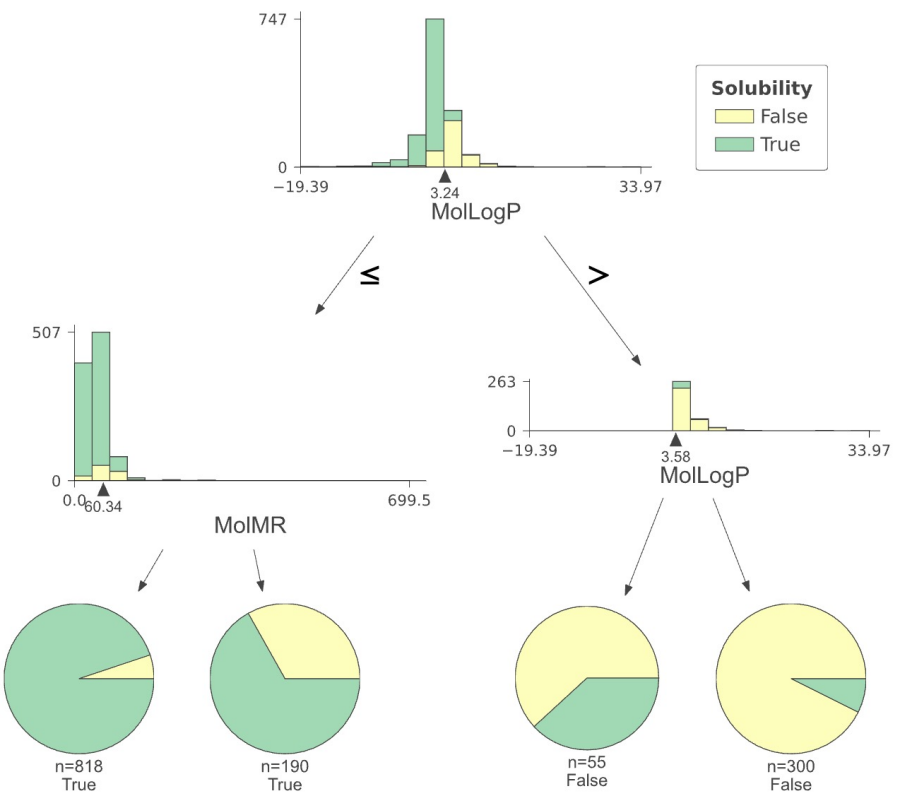
- MolWt
- MolLogP
- MolMR
- HeavyAtomCount
- NumHAcceptors
- NumHDonors
- NumHeteroatoms
- NumRotatableBonds
- NumValenceElectrons
- NumAromaticRings
- NumSaturatedRings
- NumAliphaticRings
- RingCount
- TPSA
- LabuteASA



# Increase the Decision Tree Depth to 2

**Molecular Descriptors**

- MolWt
- MolLogP
- MolMR
- HeavyAtomCount
- NumHAcceptors
- NumHDonors
- NumHeteroatoms
- NumRotatableBonds
- NumValenceElectrons
- NumAromaticRings
- NumSaturatedRings
- NumAliphaticRings
- RingCount
- TPSA
- LabuteASA



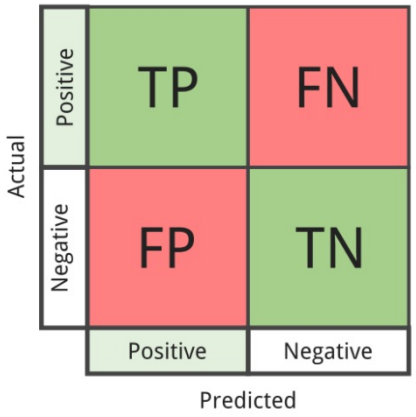
## Select Decision Tree Splits Using Gini Impurity

$$G = 1 - \sum_{k=1}^{n} {p_k}^2$$

**Gini impurity**

**0.00** **0.18** **0.24** **0.40** **0.50**

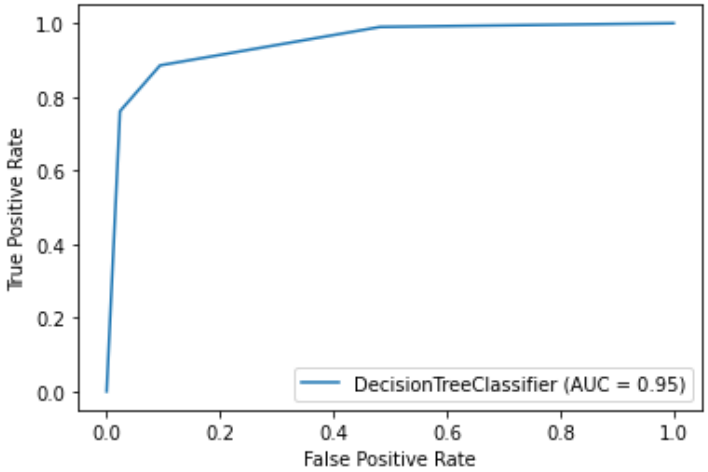
# Evaluating Classification Models




Confusion Matrix

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + FN}$$




ROC AUC Curve

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Matthews Correlation Coefficient

$$\kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)}$$

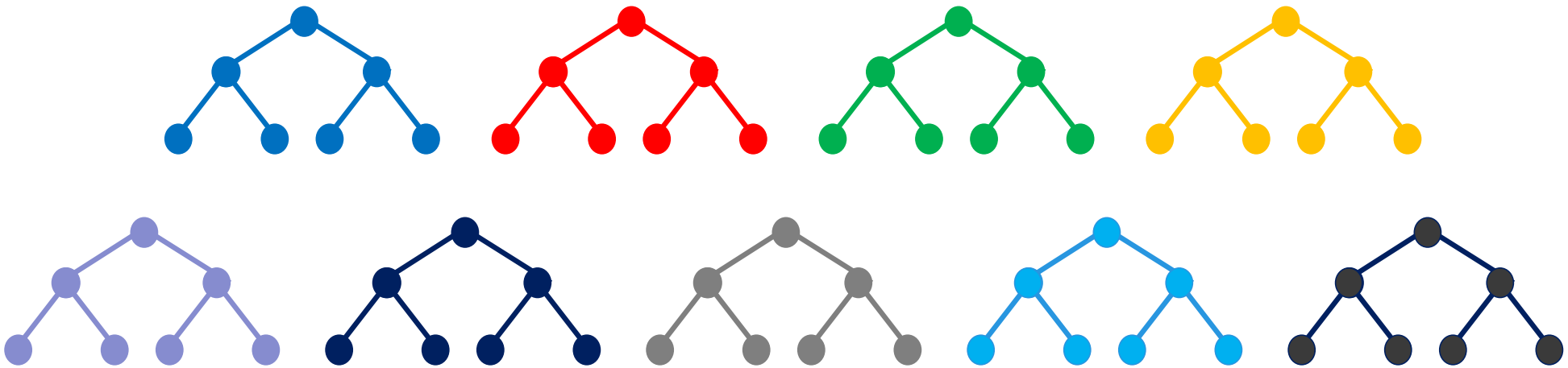
Cohen's Kappa

# Hands-on Part 2

# Ensemble models

**Random Forest**
**Extreme Gradient Boosting (XGBoost)**
**CatBoost**
**Light Gradient Boosting Machines (LightGBM)**

**Tend to work well on tabular data**



# Building A Regression Model With SciKit Learn

```
train, test = train_test_split(df)
train_X = train[descriptors]
train_y = train.experiment
test_X = test[descriptors]
test_y = test.experiment
```

Construct training and test sets

```
my_model = RandomForestRegressor()
my_model.fit(train_X, train_y)
pred = my_model.predict(test_X)
```

Build a predictive model

```
r2 = r2_score(test_y,pred)
rmse = mean_squared_error(test_y,pred,
squared=False)
```

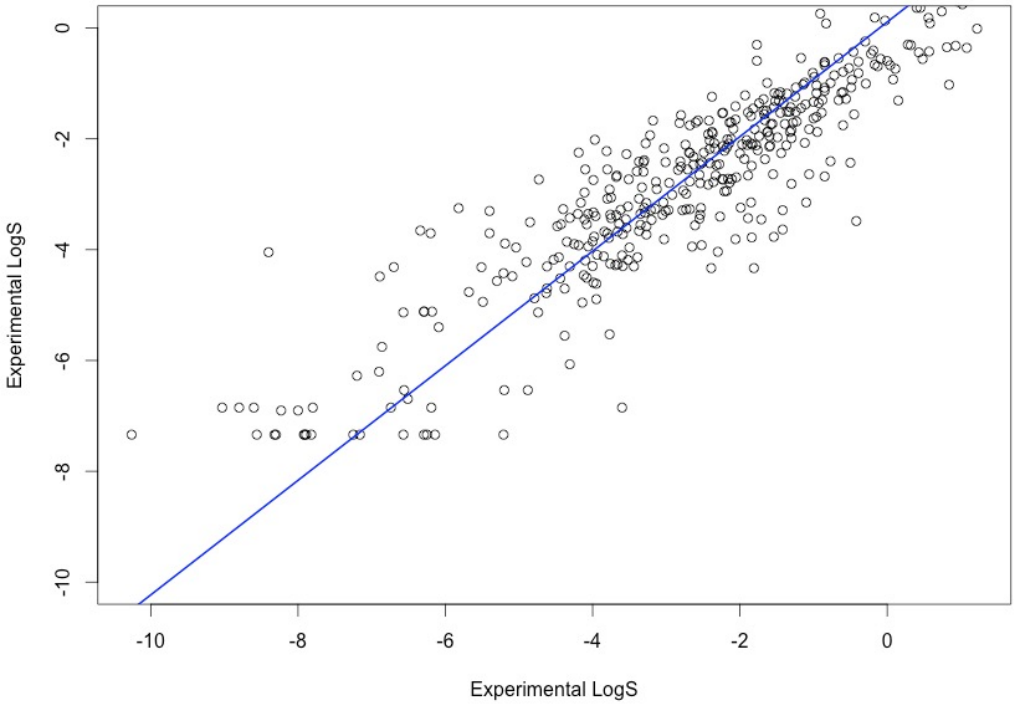
Evaluate the predictive model

# Hands-on Part 3

# Evaluating Correlation

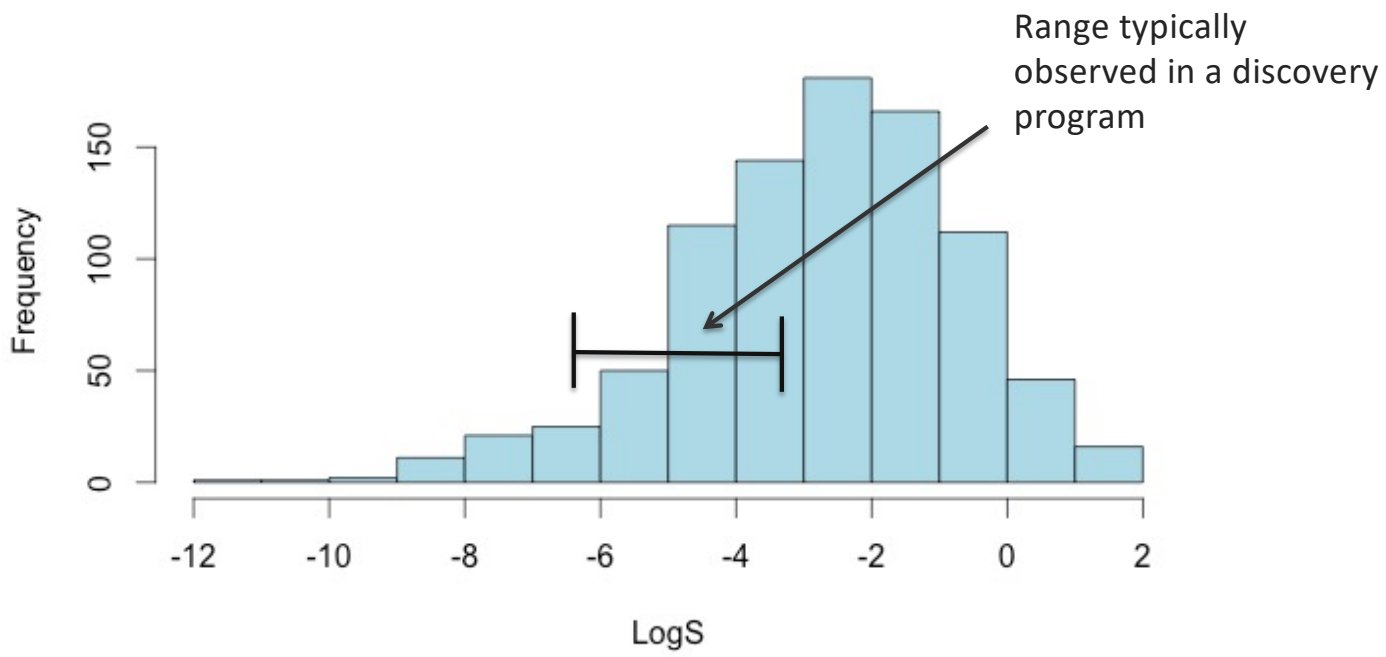
$R^2$=0.8

Median Absolute Error =0.54



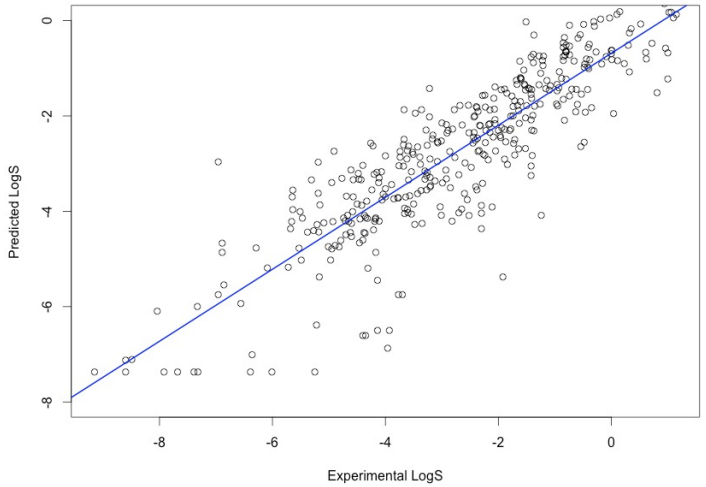
## Let's Understand the Data

- This data spans a very wide range
- Much wider than typically observed in a drug discovery program

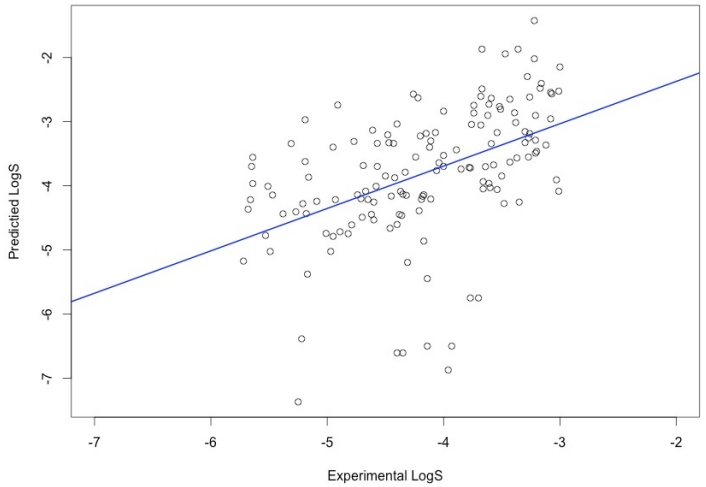


# Correlation Changes With Dynamic Range

This is the same dataset. On the left we consider the entire set, which has an unrealistically large (~10 log) dynamic range. On the right we consider a more realistic subset with a 3 log dynamic range. Note the change in correlation.

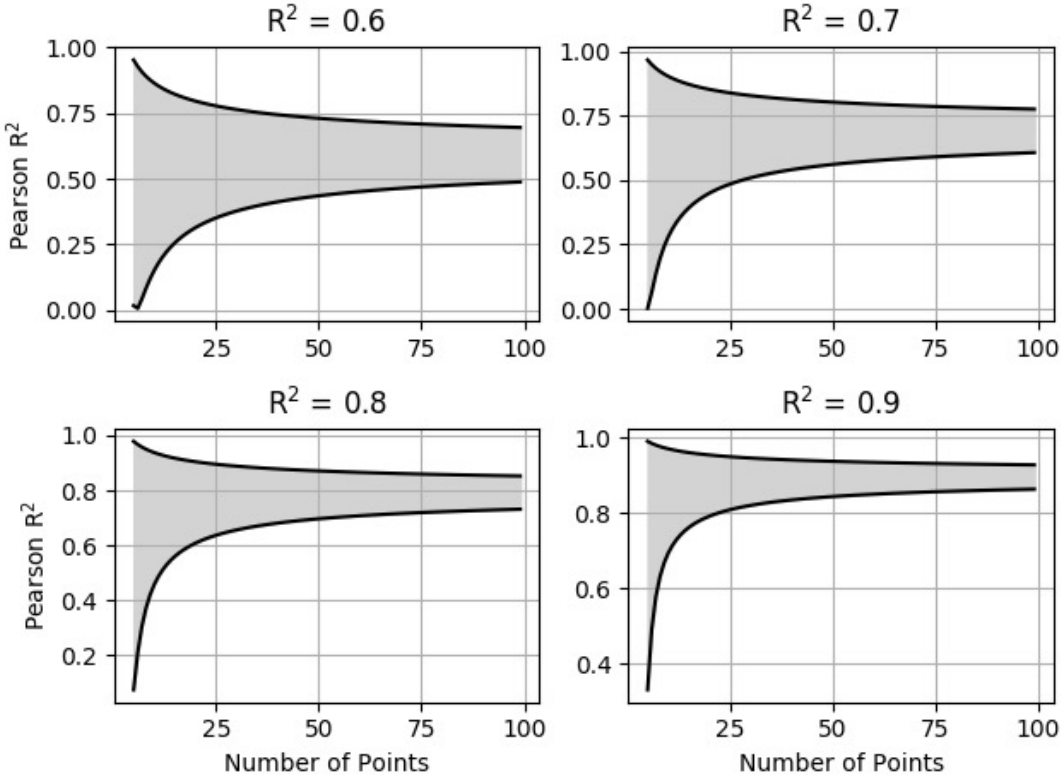



$R^2$=0.76
MAE=0.55




$R^2$=0.22
MAE=0.69

# Remember That Correlation Have Confidence Limits



## https://pingouin-stats.org


**Pingouin** is an open-source statistical package written in Python 3 and based mostly on Pandas and NumPy. Some of its main features are listed below. For a full list of available functions, please refer to the API documentation.

1. ANOVAs: N-ways, repeated measures, mixed, ancova
2. Pairwise post-hocs tests (parametric and non-parametric) and pairwise correlations
3. Robust, partial, distance and repeated measures correlations
4. Linear/logistic regression and mediation analysis
5. Bayes Factors
6. Multivariate tests
7. Reliability and consistency
8. Effect sizes and power analysis
9. Parametric/bootstrapped confidence intervals around an effect size or a correlation coefficient
10. Circular statistics
11. Chi-squared tests
12. Plotting: Bland-Altman plot, Q-Q plot, paired plot, robust correlation…